US-PAT-NO: 6104696

DOCUMENT-IDENTIFIER: US 6104696 A

TITLE: Method for sending packets between trunk ports of network switches

---------- KWIC ---------

Detailed Description Text - DETX (9):
CMIC 40 acts as a gateway between the SOC 10 and the host CPU. The communication can be, for example, along a PCI bus, or other acceptable communications bus. CMIC 40 can provide sequential direct mapped accesses between the host CPU 52 and the SOC 10. CPU 52, through the CMIC 40, will be able to access numerous resources on SOC 10, including MIB counters, programmable registers, status and control registers, configuration registers, ARL tables, port-based VLAN tables, IEEE 802.1q VLAN tables, layer three tables, rules tables, CBP address and data memory, as well as GBP address and data memory. Optionally, the CMIC 40 can include DMA support, DMA chaining and scatter-gather, as well as master and target PCI64.

Detailed Description Text - DETX (14):
The S or sideband channel runs at 132 MHz, and is 32 bits wide. The S-channel is used for functions such as four conveying Port Link Status, receive port full, port statistics, ARL table synchronization, memory and register access to CPU and other CPU management functions, and global memory full and common memory full notification.

Detailed Description Text - DETX (48):
   The S channel 83 of CPS channel 80 provides a system wide communication path
for transmitting system messages, for example, providing the CPU 52 with access
to the control structure of the SOC 10. System messages include port status
information, including port link status, receive port full, and port
statistics, ARL table 22 synchronization, CPU 52 access to GBP 60 and CBP 50
memory buffers and SOC 10 control registers, and memory full notification
corresponding to GBP 60 and/or CBP 50.

Detailed Description Text - DETX (78):
   CBM 71, in summary, performs the functions of on-chip FAP (free address
pool) management, transfer of cells to CBP 50, packet assembly and notification
to the respective egress managers, rerouting of packets to GBP 60 via a global
buffer manager, as well as handling packet flow from the GBP 60 to CBP 50.
Memory clean up, memory budget management, channel interface, and cell pointer
assignment are also functions of CBM 71. With respect to the free address
pool, CBM 71 manages the free address pool and assigns free cell pointers to
incoming cells. The free address pool is also written back by CBM 71, such
that the released cell pointers from various egress managers 76 are
appropriately cleared. Assuming that there is enough space available in CBP
50, and enough free address pointers available, CBM 71 maintains at least two
cell pointers per egress manager 76 which is being managed. The first cell of
a packet arrives at an egress manager 76, and CBM 71 writes this cell to the
CBM memory allocation at the address pointed to by the first pointer. In the
next cell header field, the second pointer is written. The format of the cell
as stored in CBP 50 is shown in FIG. 11; each line is 18

bytes wide. Line 0
contains appropriate information with respect to first cell and last cell
information, broadcast/multicast, number of egress ports for broadcast or
multicast, cell length regarding the number of valid bytes in the cell, the
next cell pointer, total cell count in the packet, and time stamp. The
remaining lines contain cell data as 64 byte cells. The free address pool
within PMMU 70 stores all free pointers for CBP 50. Each pointer in the free
address pool points to a 64-byte cell in CBP 50; the actual cell stored in the
CBP is a total of 72 bytes, with 64 bytes being byte data, and 8 bytes of
control information. Functions such as HOL blocking high and low watermarks,
out queue budget registers, CPID assignment, and other functions are handled in
CBM 71, as explained herein.

Detailed Description Text - DETX (80):
Since CBM 71 controls data flow within SOC 10, the data flow associated with
any ingress port can likewise be controlled. When packet 112 has been received
and stored in CBP 50, a CPID is provided to the associated egress manager 76.
The total number of data cells associated with the data packet is stored in a
budget register (not shown). As more data packets 112 are received and
designated to be sent to the same egress manager 76, the value of the budget
register corresponding to the associated egress manager 76 is incremented by
the number of data cells 112a, 112b of the new data cells received. The budget
register therefore dynamically represents the total number of cells designated
to be sent by any specific egress port on an EPIC 20. CBM 71 controls the
inflow of additional data packets by comparing the budget register to a high
watermark register value or a low watermark register value,

for the same
egress.

Detailed Description Text - DETX (81):
   When the value of the budget register exceeds the high watermark value, the associated ingress port is disabled. Similarly, when data cells of an egress manager 76 are sent via the egress port, and the corresponding budget register decreases to a value below the low watermark value, the ingress port is once again enabled. When egress manager 76 initiates the transmission of packet 112, egress manager 76 notifies CBM 71, which then decrements the budget register value by the number of data cells which are transmitted. The specific high watermark values and low watermark values can be programmed by the user via CPU 52. This gives the user control over the data flow of any port on any EPIC 20 or GPIC 30.

Detailed Description Text - DETX (82):
   Egress manager 76 is also capable of controlling data flow. Each egress manager 76 is provided with the capability to keep track of packet identification information in a packet pointer budget register; as a new pointer is received by egress manager 76, the associated packet pointer budget register is incremented. As egress manager 76 sends out a data packet 112, the packet pointer budget register is decremented. When a storage limit assigned to the register is reached, corresponding to a full packet identification pool, a notification message is sent to all ingress ports of the SOC 10, indicating that the destination egress port controlled by that egress manager 76 is unavailable. When the packet pointer budget register is decremented below the packet pool high watermark value, a notification message is

sent that the
destination egress port is now available. The notification messages are sent
by CBM 71 on the S channel 83.

Detailed Description Text - DETX (83):
As noted previously, flow control may be provided by CBM 71, and also by
ingress submodule 14 of either an EPIC 20 or GPIC 30. Ingress submodule 14
monitors cell transmission into ingress port 24. When a data packet 112 is
received at an ingress port 24, the ingress submodule 14 increments a received
budget register by the cell count of the incoming data packet. When a data
packet 112 is sent, the corresponding ingress 14 decrements the received budget
register by the cell count of the outgoing data packet 112. The budget
register 72 is decremented by ingress 14 in response to a decrement cell count
message initiated by CBM 71, when a data packet 112 is successfully transmitted
from CBP 50.

Detailed Description Text - DETX (124):
FFP 141 is essentially a state machine driven programmable rules engine.
The filters used by the FFP are 64 (sixty-four) bytes wide, and are applied on
an incoming packet; any offset can be used, however, a preferred embodiment
uses an offset of zero, and therefore operates on the first 64 bytes, or 512
bits, of a packet. The actions taken by the filter are tag insertion, priority
mapping, TOS tag insertion, sending of the packet to the CPU, dropping of the
packet, forwarding of the packet to an egress port, and sending the packet to a
mirrored port. The filters utilized by FFP 141 are defined by rules table 22.
Rules table 22 is completely programmable by CPU 52, through CMIC 40. The
rules table can be, for example, 256 entries deep, and may

be partitioned for
inclusive and exclusive filters, with, again as an example, 128 entries for
inclusive filters and 128 entries for exclusive filters. A filter database,
within FFP 141, includes a number of inclusive mask registers and exclusive
mask registers, such that the filters are formed based upon the rules in rules
table 22, and the filters therefore essentially form a 64 byte wide mask or bit
map which is applied on the incoming packet. If the filter is designated as an
exclusive filter, the filter will exclude all packets unless there is a match.
In other words, the exclusive filter allows a packet to go through the
forwarding process only if there is a filter match. If there is no filter
match, the packet is dropped. In an inclusive filter, if there is no match, no
action is taken but the packet is not dropped. Action on an exclusive filter
requires an exact match of all filter fields. If there is an exact match with
an exclusive filter, therefore, action is taken as specified in the action
field; the actions which may be taken, are discussed above. If there is no
full match or exact of all of the filter fields, but there is a partial match,
then the packet is dropped. A partial match is defined as either a match on
the ingress field, egress field, or filter select fields. If there is neither
a full match nor a partial match with the packet and the exclusive filter, then
no action is taken and the packet proceeds through the forwarding process. The
FFP configuration, taking action based upon the first 64 bytes of a packet,
enhances the handling of real time traffic since packets can be filtered and
action can be taken on the fly. Without an FFP according to the invention, the
packet would need to be transferred to the CPU for appropriate action to be
interpreted and taken. For inclusive filters, if there is

a filter match,
action is taken, and if there is no filter match, no action is taken; however,
packets are not dropped based on a match or no match situation for inclusive
filters.

Detailed Description Text - DETX (129):
As mentioned previously, FFP 141 is programmed by the user, through CPU 52,
based upon the specific functions which are sought to be handled by each FFP
141. Referring to FIG. 17, it can be seen that in step 17-1, an FFP
programming step is initiated by the user. Once programming has been
initiated, the user identifies the protocol fields of the packet which are to
be of interest for the filter, in step 17-2. In step 17-3, the packet type and
filter conditions are determined, and in step 17-4, a filter mask is
constructed based upon the identified packet type, and the desired filter
conditions. The filter mask is essentially a bit map which is applied or ANDed
with selected fields of the packet. After the filter mask is constructed, it
is then determined whether the filter will be an inclusive or exclusive filter,
depending upon the problems which are sought to be solved, the packets which
are sought to be forwarded, actions sought to be taken, etc. In step 17-6, it
is determined whether or not the filter is on the ingress port, and in step
17-7, it is determined whether or not the filter is on the egress port. If the
filter is on the ingress port, an ingress port mask is used in step 17-8. If
it is determined that the filter will be on the egress port, then an egress
mask is used in step 17-9. Based upon these steps, a rules table entry for
rules tables 22 is then constructed, and the entry or entries are placed into
the appropriate rules table (steps 17-10 and 17-11). These

steps are taken
through the user inputting particular sets of rules and
information into CPU 52
by an appropriate input device, and CPU 52 taking the
appropriate action with
respect to creating the filters, through CMIC 40 and the
appropriate ingress or
egress submodules on an appropriate EPIC module 20 or GPIC
module 30.

Detailed Description Text - DETX (172):

SOC 10 incorporates some unique data flow
characteristics, in order maximize
efficiency and switching speed. In network communications,
a concept known as
head-of-line or HOL blocking occurs when a port is
attempting to send a packet
to a congested port, and immediately behind that packet is
another packet which
is intended to be sent to an un-congested port. The
congestion at the
destination port of the first packet would result in delay
of the transfer of
the second packet to the un-congested port. Each EPIC 20
and GPIC 30 within
SOC 10 includes a unique HOL blocking mechanism in order to
maximize throughput
and minimize the negative effects that a single congested
port would have on
traffic going to un-congested ports. For example, if a
port on a GPIC 30, with
a data rate of, for example, 1000 megabits per second is
attempting to send
data to another port 24a on EPIC 20a, port 24a would
immediately be congested.
Each port on each GPIC 30 and EPIC 20 is programmed by CPU
52 to have a high
watermark and a low watermark per port per class of service
(COS), with respect
to buffer space within CBP 50. The fact that the head of
line blocking
mechanism enables per port per COS head of line blocking
prevention enables a
more efficient data flow than that which is known in the
art. When the output
queue for a particular port hits the preprogrammed high
watermark within the

allocated buffer in CBP 50, PMMU 70 sends, on S channel 83, a COS queue status notification to the appropriate ingress module of the appropriate GPIC 30 or EPIC 20. When the message is received, the active port register corresponding to the COS indicated in the message is updated. If the port bit for that particular port is set to zero, then the ingress is configured to drop all packets going to that port. Although the dropped packets will have a negative effect on communication to the congested port, the dropping of the packets destined for congested ports enables packets going to un-congested ports to be expeditiously forwarded thereto. When the output queue goes below the preprogrammed low watermark, PMMU 70 sends a COS queue status notification message on the sideband channel with the bit set for the port. When the ingress gets this message, the bit corresponding to the port in the active port register for the module can send the packet to the appropriate output queue.
By waiting until the output queue goes below the low watermark before re-activating the port, a hysteresis is built into the system to prevent constant activation and deactivation of the port based upon the forwarding of only one packet, or a small number of packets. It should be noted that every module has an active port register. As an example, each COS per port may have four registers for storing the high watermark and the low watermark; these registers can store data in terms of number of cells on the output queue, or in terms of number of packets on the output queue. In the case of a unicast message, the packet is merely dropped; in the case of multicast or broadcast messages, the message is dropped with respect to congested ports, but forwarded to uncongested ports. PMMU 70 includes all logic required to implement this

mechanism to prevent HOL blocking, with respect to budgeting of cells and packets. PMMU 70 includes an HOL blocking marker register to implement the mechanism based upon cells. If the local cell count plus the global cell count for a particular egress port exceeds the HOL blocking marker register value, then PMMU 70 sends the HOL status notification message. PMMU 70 can also implement an early HOL notification, through the use of a bit in the PMMU configuration register which is referred to as a Use Advanced Warning Bit. If this bit is set, the PMMU 70 sends the HOL notification message if the local cell count plus the global cell count plus 121 is greater than the value in the HOL blocking marker register. 121 is the number of cells in a jumbo frame.

Detailed Description Text - DETX (173):
With respect to the hysteresis discussed above, it should be noted that PMMU 70 implements both a spatial and a temporal hysteresis. When the local cell count plus global cell count value goes below the value in the HOL blocking marker register, then a poaching timer value from a PMMU configuration register is used to load into a counter. The counter is decremented every 32 clock cycles. When the counter reaches 0, PMMU 70 sends the HOL status message with the new port bit map. The bit corresponding to the egress port is reset to 0, to indicate that there is no more HOL blocking on the egress port. In order to carry on HOL blocking prevention based upon packets, a skid mark value is defined in the PMMU configuration register. If the number of transaction queue entries plus the skid mark value is greater than the maximum transaction queue size per COS, then PMMU 70 sends the COS queue status message on the S channel.
Once the ingress port receives this message, the ingress

port will stop sending
packets for this particular port and COS combination.
Depending upon the
configuration and the packet length received for the egress
port, either the
head of line blocking for the cell high watermark or the
head of line blocking
for the packet high watermark may be reached first. This
configuration,
therefore, works to prevent either a small series of very
large packets or a
large series of very small packets from creating HOL
blocking problems.

Detailed Description Text - DETX (176):
To summarize, resolved packets are placed on C channel
81 by ingress
submodule 14 as discussed with respect to FIG. 8. CBM 71
interfaces with the
CPS channel, and every time there is a cell/packet
addressed to an egress port,
CBM 71 assigns cell pointers, and manages the linked list.
A plurality of
concurrent reassembly engines are provided, with one
reassembly engine for each
egress manager 76, and tracks the frame status. Once a
plurality of cells
representing a packet is fully written into CBP 50, CBM 71
sends out CPIDs to
the respective egress managers, as discussed above. The
CPIDs point to the
first cell of the packet in the CBP; packet flow is then
controlled by egress
managers 76 to transaction MACs 140 once the CPID/GPID
assignment is completed
by CBM 71. The budget register (not shown) of the
respective egress manager 76
is appropriately decremented by the number of cells
associated with the egress,
after the complete packet is written into the CBP 50. EGM
76 writes the
appropriate PIDs into its transaction FIFO. Since there
are multiple classes
of service (COSs), then the egress manager 76 writes the
PIDs into the selected
transaction FIFO corresponding to the selected COS. As
will be discussed below

with respect to FIG. 13, each egress manager 76 has its own scheduler interfacing to the transaction pool or transaction FIFO on one side, and the packet pool or packet FIFO on the other side. The transaction FIFO includes all PIDs, and the packet pool or packet FIFO includes only CPIDs. The packet FIFO interfaces to the transaction FIFO, and initiates transmission based upon requests from the transmission MAC. Once transmission is started, data is read from CBP 50 one cell at a time, based upon transaction FIFO requests.

Detailed Description Text - DETX (188):
CPU 52 is treated by SOC 10 as any other port. Therefore, CMIC 40 must provide necessary port functions much like other port functions defined above. CMIC 40 supports all S channel commands and messages, thereby enabling CPU 52 to access the entire packet memory and register set; this also enables CPU 52 to issue insert and delete entries into ARL/L3 tables, issue initialize CFAP/SFAP commands, read/write memory commands and ACKs, read/write register command and ACKs, etc. Internal to SOC 10, CMIC 40 interfaces to C channel 81, P channel 82, and S channel 83, and is capable of acting as an S channel master as well as S channel slave. To this end, CPU 52 must read or write 32-bit D words. For ARL table insertion and deletion, CMIC 40 supports buffering of four insert/delete messages which can be polled or interrupt driven. ARL messages can also be placed directly into CPU memory through a DMA access using an ARL DMA controller 161. DMA controller 161 can interrupt CPU 52 after transfer of any ARL message, or when all the requested ARL packets have been placed into CPU memory.

Detailed Description Text - DETX (189):
   Communication between CMIC 40 and C channel 81/P channel
82 is performed
through the use of CP-channel buffers 162 for buffering C
and P channel
messages, and CP bus interface 163. S channel ARL message
buffers 164 and S
channel bus interface 165 enable communication with S
channel 83. As noted
previously, PIO (Programmed Input/Output) registers are
used, as illustrated by
SCH PIO registers 166 and PIO registers 168, to access the
S channel, as well
as to program other control, status, address, and data
registers. PIO
registers 168 communicate with CMIC bus 167 through 12C
slave interface 42a and
12C master interface 42b. DMA controller 161 enables
chaining, in memory,
thereby allowing CPU 52 to transfer multiple packets of
data without continuous
CPU intervention. Each DMA channel can therefore be
programmed to perform a
read or write DMA operation. Specific descriptor formats
may be selected as
appropriate to execute a desired DMA function according to
application rules.
For receiving cells from PMMU 70 for transfer to memory, if
appropriate, CMIC
40 acts as an egress port, and follows egress protocol as
discussed previously.
For transferring cells to PMMU 70, CMIC 40 acts as an
ingress port, and follows
ingress protocol as discussed previously. CMIC 40 checks
for active ports, COS
queue availability and other ingress functions, as well as
supporting the HOL
blocking mechanism discussed above. CMIC 40 supports
single and burst PIO
operations; however, burst should be limited to S channel
buffers and ARL
insert/delete message buffers. Referring once again to 12C
slave interface
42a, the CMIC 40 is configured to have an 12C slave address
so that an external
12C master can access registers of CMIC 40. CMIC 40 can
inversely operate as
an 12C master, and therefore, access other 12C slaves. It

should be noted that
CMIC 40 can also support MIIM through MIIM interface 169.
MIIM support is
defined by IEEE Standard 802.3u, and will not be further
discussed herein.
Similarly, other operational aspects of CMIC 40 are outside
of the scope of
this invention.

Current US Cross Reference Classification - CCXR (2):
370/225